UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/204,973	12/03/98	EHNEBUSKE D	AT9-98-266

TM02/1002

DUKE W YEE
CARSTENS YEE AND CAHOON
P O BOX 802334
DALLAS TX 75380

EXAMINER	
INGBERG, T	
ART UNIT	**PAPER NUMBER**
2122	

DATE MAILED: 10/02/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary

Application No.	Applicant(s)
09/204,973	David Lars Ehnebuske et al.
Examiner	**Art Unit**
Todd Ingberg	2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *Feb 5, 1999* .

2a) ☐ This action is **FINAL.** 2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-98* is/are pending in the application.

4a) Of the above, claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *1-98* is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claims _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

11) ☒ The proposed drawing correction filed on *Mar 15, 1999* is: a) ☒ approved b) ☐ disapproved.

12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

13) ☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

a) ☐ All b) ☐ Some* c) ☐ None of:

1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

14) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

15) ☒ Notice of References Cited (PTO-892)
16) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
17) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s). _____
18) ☒ Interview Summary (PTO-413) Paper No(s). 4
19) ☐ Notice of Informal Patent Application (PTO-152)
20) ☐ Other:

DETAILED ACTION

Claims 1 - 98 have been examined.

Claims 12, 18, 19, 20 , 21, 30, 54, 70 71, 72, 76, 77, 78,80 and 81 were amended in a preliminary amendment received February 5, 1999.

Drawings

1. The corrected or substitute drawings were received on March 15, 1999. These drawings are approved by the Draft's Person.

Information Disclosure Statement

2. In the event the invention is related to the Assignee's (IBM) product **FLOWMARK** (IBM Trademark #2006543) the Applicant is reminded of their duty to disclose. FLOWMARK dates back to filing for Trademark on December 16, 1993. Date of first use in commerce June 28, 1995.

3. The Applicant makes specific reference to Object Management Group (OMG), the Assignee (IBM) is a long time member of this standards organization for Object technology. In the event work of OMG is relevant to Common Object Request Broker (**CORBA**), application frameworks and Business Objects, the Applicant is reminded of their duty to disclose.

4. In the event, the invention is related to the tool **ISMOD** used by IBM in 1990 to model dimensions of business such as "data and information, data criticism, data flow identification, data flow metrics, data qualifiers, event triggers, location, organization, processes", as described in

IBM System Journal, Vol 29 Issue 4, page 509, page 17, then, the Applicant is reminded of their duty to disclose.

5. In the event, the invention is related to the tool **DevelopMate** used by IBM in 1990 to develop a business/enterprise model with dimensions that include business processes, business data events ..." , as described in IBM System Journal, Vol 29 Issue 4, page 509, page 17, then, the Applicant is reminded of their duty to disclose.

6. In the event, the invention is related to the product of **Newi** from Integrated Objects - a joint venture of IBM and Softwright based in England, the Applicant is reminded of their duty to disclose.

7. In the event, the invention is related to feature "rule editor probe", contained in the product **Object Management Workbench** (OMW) of Intellicorp, the Applicant is reminded of their duty to disclose.

Currently, compliance with this request is only covered under 1.56 but could be the subject of a future Requirement For Information (RFI) under 1.105.

Specification

8. The Specification contains and error on page 6. No case with the Serial number of 09/993,718 has been filed with the USPTO at this time. The correct serial number is needed as part of an amendment.

Examiner's Interpretation

9. The following are Examiner interpretations.

a. ***Modeling*** - The Martin book teaches the use of modeling an enterprise (**Martin**, page 247 - 249) operation and provides Appendix A, Recommended Diagraming Standards. Martin page 285 illustrates the transition from the problem domain, to modeling, to OO Design to code. The Martin reference has many chapters covering modeling and discusses the models are tied together to generate code (**Martin,** page 155, box).

The Martin reference also provides some examples. These models are not viewed as static. The Martin reference teaches modeling the enterprise. The principles and techniques are dynamic.

b. ***Flow Control*** - The Applicant does not explicitly claim flow control. However, when the Applicant states in the claims "method logic is continuous" this is interpreted as meaning the method (a common feature in object technology) can run until a outside interrupt occurs. This is a product of flow control resulting from the logic structure of a computer program. There are many claims to what the Examiner interprets as claims to flow control. Flow control is the tracing the path of an executing of a program. The exact path the execution of a program will follow is determined by the values of the attributes and the control conditions encountered. The examples of the programming constructs such as Martin page 148 show the difference paths flow control can take depending on execution of operations such as CHECK IN COPY , FILL REQUEST and BOOK OVERDUE. The values of attributes are tested to determine the path taken. Many claims have made claim to flow control which is inherent to the execution of a computer programs.

c. Claim 3 contains the following limitation ".... the step of defining a first control point further comprises:

(a1) decorating the object to dynamically insert a first control point such that the object acquires this new control point."

The Examiner interprets the "decorating" to mean the entry of programming information such as the operations/methods and the entry of control points in a programming environment.

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C.§ 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. Claims 1 - 98 are rejected under 35 U.S.C. 102(a) as being anticipated by "Principles of Object-Oriented Analysis and Design", James **Martin**, published June 1, 1992.

The Martin teaches the underlying theory of building an Object Oriented Computer Aided Software Engineering (OO-CASE) tools in his 1992 text book. The Martin should be taken as a whole, however, focus of the rejection is on the Chapters 9 and 10. The Martin references covers the very basics of object technology that one of ordinary skill should have known well before the time of invention:

Chapter 2 - Basic Concepts

Chapter 3 - Why Object-Oriented ?

Chapter 4 - Basic Guidelines

Chapter 6 - Categorizing Objects

Chapter 7 - Relationships Among Object Types

Chapter 8 - State and State Changes

Chapter 9 - Events, Triggers, and Operations

Chapter 10 - Rules

Chapter 11 - How Diagrams Interrelate

Chapter 12 - Basic Concepts of OO Design

Chapter 15 - Method Creation

Chapter 18 - OO-CASE Tools

Appendix A - Recommended Diagraming Standards

The Martin book in addition to containing foundation knowledge of object oriented technology it teaches applying a set of rules comprising the placing of logic (program statements) in a pre-method control before the logic of a method and post method control point after the logic of a method. Martin also teaches associating a set of rules with each control point based on the class of the object in which the method resides, name of the method and type of control point and invoking methods.

Claim 1

Martin anticipates a computer implemented process for applying a set of rules (**Martin,** Chapter 10, RULES, and page 138-139 and 249-251), the process comprising:

(a) placing a pre-method control before logic of a method (**Martin,** page 142, operation precondition) and post-method control point after the logic of the method (**Martin,** page 142, postcondition)

(b) associating a set rules with each control point (**Martin,** page 142, 147 "Operation" as per (a) above) based on a class of object in which the method resides (**Martin,** page 143, "... rules associated with diagrams of OO ...") , name of the method, and type of control point, whether the pre-method control point or the post-method control point (**Martin,** page 142, operation precondition) ;

(c) invoking the method (**Martin**, page 116), wherein encountering each control point during the execution of the method comprises (**Martin,** page 142, postcondition):

(i) determining if the encountered control point is active (**Martin,** page 122, IF structure in center diagram) ;

(ii) on the basis of an active control point (Interpreted as the result of the IF structure above further described in Appendix A on page 381 Control Conditions):

1) selecting rules based on a set of rules associated with the active control point associated in step (**Martin,** page 122, first diagram example is the control condition to fire missile) (b);

2) running the selected rules (**Martin,** page 122, rule that lead to the control condition);

3) obtaining results from running the rules (**Martin,** page 122, trigger rule at the bottom of the page); and

4) combining the results using a combining algorithm specified by the control point (**Martin,** page 122, A control condition can function as a combining algorithm as seen in diagram in middle of the page and page 126 Figure 9.9 and **Martin** teaches a way to have a combing algorithm where one of three operations are selected as on page 124, and **Martin** teaches a way to have a combining algorithm where one can be selected as taught in the mutually exclusive notation on the bottom of page 125).

Claim 2

Martin anticipates a computer implemented process for applying a set of rules (**Martin**, Page 166, Rules Linked to Diagrams a product and **Martin**, page 172 from operations to methods) comprising:

(a) defining an object (**Martin,** page 171, CLASS - bottom of page);

(b) defining at least one method in the object (**Martin**, page 173, Method - top of page and page 116 and page 167 Rule editor);

(c) defining a control point just before logic of at least one method (**Martin**, page 173, diagram in center of page); and

(d) associating a set of rules with the control point (**Martin**, page 173, diagram in center of page).

Claim 3

In the process of claim 2, the step of defining a first control point further comprises:

(a1) decorating the object to dynamically insert a first control point such that the object acquires this new control point (**Martin,** teaches how the object model and rules are associated in Chapter 5 page 59, on page 60 of Martin last paragraph OO-CASE tools states last paragraph "Every time we tell an advanced CASE tool about classes, inheritance, and so on, it should generate code." The next sentence mentions code can be generated from rules. The reference also mentions the diagrams are executable such as on page 142 - The reference teaches Object Oriented Computer Aided Software Engineering (OO-CASE) tools and their features. OO-CASE tools as described create objects and methods much of which is described in claim 2).

Claim 4

In the process of claim 2, the step of defining at least one control point further comprises:

(c1) adding the at least one control point through the technique of generating required code in the compiler or with a preprocessor. As per claim 3 the diagrams generate code also see the definition of Instant CASE or I-CASE **Martin,** pages 7, 243, 282-283, 284, 293, 294, 351 and 352.

Claim 5

In the process of claim 2, the step of defining at least one control point further comprises:

(c1) manually inserting the at least one control point and encoding the control point in the implementation of a hosting object. As per claim 3.

Claim 6

In the process of claim 2, the step of defining at least one control point further comprises:

(cl) externalizing the at least one control point as a class and instantiating it at the at least one control point (**Martin**, page 133 - 136, BOX and claim 1 and page 167 OMW screen shot)

Claim 7

The process of claim 2 further comprises:

(e) defining a second control point just after the logic of each method; and

(f) associating a second set of rules with the second control point (**Martin,** in many locations teaches the ability to have more than 1 control point and additional rules see page 164 for an example).

Claim 8

In the process of claim 7, wherein the rules in the second set of rules are associated to the second control point without considering the rules in the first set of rules associated with the at least one control point as per claim 7.

Claim 9

In the process of claim 7, wherein a set of rules is defined as having N number of rules, N being at least zero. As per claim 1 the presents of a single RULE means it exists and the count of rules present is greater than zero.

Claim 10

In the process of claim 2, the step of associating at least one control point further comprises:

(cl) defining, with a control point, at least one of a rule selecting algorithm and a rule-results combination algorithm As per claim 1.

Claim 11

The process of claim 2, further comprises:

(e) changing rules associated with the control point contained in the set of rules. As per claims 3 and claim 4 OO-CASE and I-CASE by definition.

Claim 12

Martin anticipates a computer implemented process for applying a set of rules (as per claim 2), comprising:

(a) invoking a method in an object (as per claim 2);

(b) encountering an active control point during the invocation of the method (as per claim 2);

(c) selecting rules associated with the method of the object at the control point (as per claim 2);

(d) invoking the rules (as per claim 2); and

(e) combining results from invoking the rules as per claim 1.

Claim 13

The process of claim 12, wherein the rules perform a variety of actions (Martin, page 164, a variety of actions can occur such as Invoice Student OR Get Dorm depending on the outcome of the Remote Student registered condition) conditioned by the fact that rules may be associated with particular, regularly occurring points in the object model Martin,

(**Martin,** page 166, RULES LINKED TO DIAGRAMS, "The importance of rules was emphasized in Chapter 10 which indicated that rules can be connected to any of the OO diagrams".)

Claim 14

The process of claim 12, wherein the rules perform at least one function which varies over time (**Martin,** page 117, clock events and page 144 Rules Associated with Event Diagrams - " If time is between 9 AM to 5 PM", **Martin**, page 394, Clock Events).

Claim 15

A process of claim 12, wherein a control point occurs just before logic of the method begins, just after the logic of the method completes, or at both just before logic of the method begins and just after the logic of the method completes as per claim 1.

Claim 16

Martin anticipates a computer implemented process for applying a set of rules (as per claim 2) comprising:

(a) defining an object (as per claim 2);

(b) defining at least one method in the object (as per claim 2);

(c) defining at least one control point in the at least one method (as per claim 2).

(d) defining rules to the at least one control point on basis the object's class name, method, name, and position of the at least one control point in the method (Martin, Chapter 12, BASIC CONCEPTS OF OO DESIGN, page 172 - 173, the relationship between classes and objects and the relationship between rules and object modeling).

Claim 17

In the process of claim 16, further comprising the step of activating at least one control point having associated rules as per claim 1.

Claim 18

The process of claim 16 further comprises:

(e) encountering a first control point (**Martin,** page 173, control point with a Time Event) ;

(f) running the rules associated with the first control point (**Martin,** page 173, control point with a Time Event); and

(g) affecting behavior of the object based on running the rules associated with the first control point (The flow control is controlled by the Rule associated with the Control point as per **Martin,** page 381).

Claim 19

In the process of claim 18, the step of affecting the behavior of the object further comprises:

(h) associating different rules to a control point (as per claim 14 - Different rules based on the time of day affects the flow control/ behavior).

Claim 20

In the process of claim 18, the step of affecting the behavior of the object further comprises:

(h) defining another control point (Examiner Interpretation of "defining another control point" the meaning could be at design time or runtime. Design time would involve the interaction with the OO-CASE tool as on **Martin,** page 162, Run time would mean the behavior changes value such as attribute which influence the path the control flow takes. This is the point of programming. The

ability to model a problem domain and execute code that process information that reflects the modeled problem - Flow Control as determined by the running of the program such as **Martin,** page 163).

Claim 21

In the process of claim 18, the step of modifying the object further comprises:

(h) associating rules to a second control point (**Martin,** page 163, Multiple control points defined).

Claim 22

In the process of claim 16, further comprising a step of deactivating the at least one control point.(As per claim 1. The control point is determined if it is active or not. If one takes the **Martin**, page 163 example where the timed event is part of the Waitlisted functionality the Timed event occurs at a specific time the Timed Event is one example of activating and deactivating the control point also see **Martin**, Appendix A, page 394)

Claim 23

Martin anticipates a computer implemented process for applying a set off rules (as per claim 2), comprising

(a) defining an object (as per claim 2);

(b) defining a method in the object (as per claim 2);

(c) defining a first control point of the method (as per claim 2);

(d) determining rules associated with the first control point (as per claim 2);

(e) defining a second control point of the method (as per claim 2); and

(f) determining rules associated with the second control point (as per claim 2).

Claim 24

A computer implemented process as in claim 23 further comprising:

(g) separately selecting, running and combining the results of rules determined to be associated with either control point as per claim 1.

Claim 25

In the process of claim 23 wherein the first control point is a pre-method trigger point (**Martin**, page 142, diagram top of page, page 381 Trigger Rules).

Claim 26

In the process of claim 23 wherein the second control point is a post-method trigger point (**Martin**, page 115, Postconditions in cause and effect isolation, page 141, Post Condition page 381, Trigger Rule).

Claim 27

Martin anticipates a computer implemented process for defining an object (Martin, page 166 - 167, Link between, Diagrams, Rules and Objects) comprising:

defining an object; (**Martin**, page 144, Box 10.3, and page 169 - 176 and as per claim 2)

defining a method in the object by: defining method logic (as per claim 2) ;

placing the method logic in the method (**Martin**, page 173, methods is the Specification of an operation and as per claim 2);

defining at least one control point (as per claim 2);

and placing the at least one control point in the method wherein the method logic is continuous. **(Martin**, page 224, DO and FOR loops, page 225, Loops in action diagrams).

Claim 28

A computer implemented process for defining an object as in claim 27, wherein the step of placing the at least one control point further comprises placing the at least one control in the method before the method logic (as per claim 1).

Claim 29

A computer implemented process for defining an object as in claim 27, wherein the step of placing the at least one control point further comprises placing the at least one control in the method after the method logic (as per claim 1).

Claim 30

A computer implemented process for defining an object as in claim 27, wherein the at least one control point comprises two control points and further comprises: placing a first control point in the method before the method logic; and placing a second control point in the method after the method logic (Martin, page 126, Figure 9.9).

Claim 31

A computer implemented process for defining an object as in claim 27, further comprises: flagging the at least one control point on the basis of being active (as per claim 1).

Claim 32

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprising: defining a rule selection algorithm associated with the at least one control point (**Martin,** page 168, control point rule illustrated).

Claim 33

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprising: defining a rule result combination algorithm associated with the at least one control point. As per claim 1.

Claim 34

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprises: defining a rule selection algorithm for the at least one control point; and defining a rule result combination algorithm for the at least one control point As per claim 1.

Claim 35

A computer implemented process for defining an object as in claim 27, further comprising: associating at least one rule with the at least one control point. As per claim 32.

Claim 36

Martin anticipates a computer implemented process for defining a rule comprising: creating the rule (**Martin**, page 167, Rule Editor) ; associating the rule with an object class (**Martin**, page 167, Figure 11.14); associating the rule with a method within the object class (Martin, page 173,

operations are methods); and associating the rule with an occurrence of a control point within the method (**Martin,** page 168, Figure 11.16).

Claim 37

A computer implemented process for defining a rule as in claim 36 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

Claim 38

A computer implemented process for defining a rule as in claim 36 wherein the occurrence of control point within the method being after method logic. As per claim 1.

Claim 39

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another object class (**Martin,** page 267, the ability to access a method/Rule from more than one object and the concept of Reuse which is a key factor in object oriented technology **Martin**, page 248, Box 16.2 Maximize reusability) (This claim could also be interpreted as claiming the principle of inheritance as described on **Martin**, page 266 - 268).

Claim 40

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another method within the object class. As per claim 39.

Claim 41

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another control point within the method of the object class (**Martin,** page 166 - 168, the rule associated to the control point , page 233, RULES)

Claim 42

Martin anticipates a computer implemented process for applying a set of rules (as per claim 2), comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules associated with the method comprising: encountering a control point associated with the method; determining if the control point is active; and finding at least one rule associated with an active control point. (Interpreted as the running of the code generated by claim 2).

Claim 43

A computer implemented process for applying a set of rules as in claim 42, wherein the step of finding at least one rule further comprises: accessing a selecting algorithm associated with the active control point (as per claim 1); and selecting at least one rule using the selecting algorithm (as per claim 10 and The IF structure in the control point as per **Martin**, page 168).

Claim 44

A computer implemented process for applying a set of rules as in claim 42, where in the step of processing rules further comprises: running the at least one rule; determining results from running the at least one rule; accessing a combining algorithm associated with the control point; and combining the results using the combining algorithm. As per claim 1.

Claim 45

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a control point; accessing a selecting algorithm associated with the control point; and selecting at least one rule using the selecting algorithm. As per claim 42.

Claim 46

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a control point; finding at least one rule associated with the control point; running the at least one rule; determining results on the basis of running the at least one rule; accessing a combining algorithm associated with the control point; and combining the results using the combining algorithm. As per claim 1 - the running of the executable generated from the model.

Claim 47

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a first control point associated with the method; determining if the first control point is active (the running of code from claims 1 and 2 and implementations such as page 164 Fig 11.10); executing method logic of the method;

encountering a second control point associated with the method; determining if the second control point is active; finding a set of rules associated with one of the first control point and the second control point, wherein the set of rules contains not less than zero rules as per claim 9.

Claim 48

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a control point associated with the method; finding at least one rule associated with the control point prior to executing method logic of the method; running the at least one rule; obtaining results on the basis of running the at least one rule; and controlling the method on the basis of the results (The running of code from claims 1 and 2).

Claim 49

A computer implemented process for applying a set of rules as in claim 48, wherein the step of controlling the method comprises: exiting the method. **Martin**, page 236, use of "return" in C++ and it is well known in C++ that reaching the end of a method such as flow control reaching the last "}" in the method declassify will return flow control to the method that called this method or terminate. In either path the method has performed an exit.

Claim 50

A computer implemented process for applying a set of rules as in claim 48, wherein the step of controlling the method comprises: executing method logic of the method. The running of the executable code produced by the modeling from claim 1 - Flow control.

Claim 51

Martin anticipates a data processing system for defining an object comprising: defining means for defining an object; defining means for defining a method in the object by: defining means for defining method logic; placing means for placing the method logic in the method; defining means for defining at least one control point; and placing means for placing the at least one control point in the method wherein the method logic is continuous. As per claim 27.

Claim 52

A data processing system for defining an object as in claim 51, wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method before the method logic. As per claim 1.

Claim 53

A data processing system for defining an object as in claim 51, wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method after the method logic. As per claim 1.

Claim 54

A data processing system for defining an object as in claim 51, wherein the at least one control point comprises two control points and further comprises: placing means for placing a first control point in the method before the method logic; and placing means for placing a second control point in the method after the method logic. As per claims 2 and 7.

Claim 55

A data processing system for defining an object as in claim 51, further comprises: flagging means for flagging the at least one control point on the basis of being active. As per claim 31.

Claim 56

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point further comprising: defining means for defining a rule selection algorithm associated with the at least one control point. As per claim 32.

Claim 57

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point further comprising: defining means for defining a rule result combination algorithm associated with the at least one control point as per claim 10.

Claim 58

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point (as per claims 1 and 2) further comprises: defining means for defining a rule selection algorithm for the at least one control point; and defining a rule result combination algorithm for the at least one control point. As per claim 34.

Claim 59

A data processing system for defining an object as in claim 51, further comprising: associating means for associating at least one rule with the at least one control point. As per claim 8.

Claim 60

Martin anticipates a data processing system for defining a rule comprising: creating means for creating the rule; associating means for associating the rule with an object class; associating means for associating the rule with a method within the object class; and associating means for associating the rule with an occurrence of a control point within the method. As per cliam 36.

Claim 61

A data processing system for defining a rule as in claim 60 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

Claim 62

A data processing system for defining a rule as in claim 60 wherein the occurrence of control point within the method being after method logic. As per claim 1.

Claim 63

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another object class. (**Martin,** page 267, the ability to access a method/Rule from more than one object and the concept of Reuse which is a key factor in object oriented technology **Martin**, page 248, Box 16.2 Maximize reusability) (This claim could also be interpreted as claiming the principle of inheritance as described on **Martin**, page 266 - 268).

Claim 64

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another method within the object class. As per claim 39.

Claim 65

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another control point within the method of the object class. As per claim 1.

Claim 66

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules associated with the method comprising: encountering means for encountering a control point associated with the method; determining means for determining if the control point is active; and finding means for finding at least one rule associated with an active control point. As per claim 42.

Claim 67

A data processing system for applying a set of rules as in claim 66, wherein the step of finding at least one rule further comprises: accessing means for accessing a selecting algorithm associated with the active control point; and selecting means for selecting at least one rule using the selecting algorithm. As per claim 43.

Claim 68

A data processing system for applying a set of rules as in claim 66, where in the step of processing rules further comprises: running means for running the at least one rule; determining means for determining results from running the at least one rule; accessing means for accessing a combining

algorithm associated with the control point; and combining means for combining the results using the combining algorithm. As per claim 44.

Claim 69

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point; accessing means for accessing a selecting algorithm associated with the control point; and selecting means for selecting at least one rule using the selecting algorithm. As per claim 45.

Claim 70

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point; finding means for finding at least one rule associated with the control point; running means for running the at least one rule; determining means for determining results on the basis of running the at least one rule; accessing means for accessing a combining algorithm associated with the control point; and combining means for combining the results using the combining algorithm. As per claim 46.

Claim 71

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a first control point associated with the method; determining means for determining if the first control point is active (as per claim 2); executing means for executing method logic of the method (as per claim 2); encountering means for encountering a second control point associated with the method; determining means for determining if the second control point is active; finding, means for finding a set of rules associated with one of the first control point and the second control point (as per claim 7), wherein the set of rules contains not less than zero rules. As per claim 9.

Claim 72

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point associated with the method; finding means for finding at least one rule associated with the control point prior to executing method logic of the method; running means for running the at least one rule; obtaining means for obtaining results on the basis of running the at least one rule; and controlling means for controlling the method on the basis of the results. As per claim 48.

Claim 73

A data processing system for applying a set of rules as in claim 72, wherein the step of controlling the method comprises: exiting means for exiting the method. As per claim 49.

Claim 74

A data processing system for applying a set of rules as in claim 72, wherein the step of controlling the method comprises: executing means for executing method logic of the method. As per claim 50.

Claim 75

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for defining an object, the instruction comprising: instructions for defining an object; instructions for defining a method in the object by: instructions for defining method logic; instructions for placing the method logic in the method; instructions for defining at least one control point; and instructions for placing the at least one control point in the method wherein the method logic is continuous. As per claim 51.

Claim 76

A computer program product for defining an object as in claim 75, wherein the instruction of placing the at least one control point further comprises placing the at least one control point in the method before the method logic. As per claim 1.

Claim 77

A computer program product for defining an object as in claim 75, wherein the instruction of placing the at least one control point further comprises placing the at least one control point in the method after the method logic. As per claim 1.

Claim 78

A computer program product for defining an object as in claim 75, wherein the at least one control point further comprises two control points and further comprises: instructions for placing a first control point in the method before the method logic; and instructions for placing a second control point in the method after the method logic.As per claim 1.

Claim 79

A computer program product for defining an object as in claim 75, further comprises: instructions for flagging the at least one control point on the basis of being active. As per claim 31.

Claim 80

A computer program product for defining an object as in claim 75, wherein the instruction of defining the at least one control point further comprising: instructions for defining a rule selection algorithm associated with the at least one control point. As per claim 32.

Claim 81

A computer product for defining an object as in claim 75, wherein the instruction of defining the at least one control point further comprises: instructions for defining a rule combination algorithm associated with the at least one control point. As per claim 33.

Claim 82

A computer program product for defining an object as in claim 75, wherein the step of defining the at least one control point further comprises: instructions for defining a rule selection algorithm for the at least one control point; and instructions for defining a rule result combination algorithm for the at least one control point.As per claim 34.

Claim 83

A computer program product for defining an object as in claim 75, further comprising: instructions for associating at least one rule with the at least one control point. As per claim 35.

Claim 84

Martin anticipates a computer program product embodied on a computer" readable medium containing instructions for a computer implemented process for defining a rule, the instruction comprising: instructions for creating the rule; instructions for associating the rule with an object class; instructions for associating the rule with a method within the object class; and instructions for associating the rule with an occurrence of a control point within the method. As per claim 36.

Claim 85

A computer program product for defining a rule as in claim 84 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

Claim 86

A computer program product for defining a rule as in claim 84 wherein the occurrence of control point within the method being after method logic.As per claim 1.

Claim 87

A computer program product for defining a rule as in claim 84, further comprising: instructions for associating the rule with another object class. As per claim 39 or 63.

Claim 88

A computer program product for defining a rule as in claim 84, further comprising: instructions for associating the rule with another method within the object class. As per claim 39 or 64.

Claim 89

A computer implemented process for defining a rule as in claim 84, further comprising: instructions for associating the rule with another control point within the method of the object class. As per claim 65.

Claim 90

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules associated with the method comprising: instructions for encountering a control point associated with the method; instructions for determining if the control point is active; and instructions for finding at least one rule associated with an active control point. As per claim 1.

Claim 91

A computer program product for applying a set of rules as in claim 90, wherein the step of finding at least one rule further comprises: instructions for accessing a selecting algorithm associated with

the active control point; and instructions for selecting at least one rule using the selecting algorithm. As per claim 43.

Claim 92

A computer program product for applying a set of rules as in claim 90, where in the step of processing rules further comprises: instructions for running the at least one rule; instructions for determining results from running the at least one rule; instructions for accessing a combining algorithm associated with the control point; and instructions for combining the results using the combining algorithm. As per claim 1.

Claim 93

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a control point; instructions for accessing a selecting algorithm associated with the control point; and instructions for selecting at least one rule using the selecting algorithm. As per claim 42.

Claim 94

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a

method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a control point; instructions for finding at least one rule associated with the control point; instructions for running the at least one rule; instructions for determining results on the basis of running the at least one rule; instructions for accessing a combining algorithm associated with the control point; and instructions for combining the results using the combining algorithm. As per claim 1 or 46.

Claim 95

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class (as per claim 42); instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a first control point associated with the method; instructions for determining if the first control point is active; instructions for executing method logic of the method; instructions for encountering a second control point associated with the method; instructions for determining if the second control point is active (Asper claim 47); instructions for finding a set of rules associated with one of the first control point and the second control point, wherein the set of rules contains not less than zero rules. As per claim 9.

Claim 96

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the

instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; processing rules comprising: instructions for encountering a control point associated with the method; instructions for finding at least one rule associated with the control point prior to executing method logic of the method; instructions for running the at least one rule; instructions for obtaining results on the basis of running the at least one rule; and instructions for controlling the method on the basis of the results. As per claim 48.

Claim 97

A computer program product for applying a set of rules as in claim 96, wherein the step of controlling the method comprises: instructions for exiting the method. **Martin**, page 236, use of "return" in C++ and it is well known in C++ that reaching the end of a method such as flow control reaching the last "}" in the method declassify will return flow control to the method that called this method or terminate. In either path the method has performed an exit.

Claim 98

A computer program product for applying a set of rules as in claim 96, wherein the step of , controlling the method comprises: instructions for executing method logic of the method. As per claims 50.

Summary

12. The rejection under Martin is a fundamental teaching from June 1992 which contains features inherent to Object Oriented technology and a primer book on the technology. The WFT

rejection is a commercial product for use and for sale since April 1997 which implementations Rules in an Object Oriented CASE tool. Both rejections support the concept that the tools are designed for the end user to implement rules without formal programming experience. Rational Rose is also a OO-CASE tool which teaches the ability implement rules at a Programmer level.

Conclusion

13. The undisclosed prior art of Assignee **(IBM)** is made of record and not relied upon is considered pertinent to applicant's disclosure.

A. "Business/Enterprise Modeling", Robert Katz, **IBM Systems Journal**, Armonk 1990, Vol 29, Issue 4, page 509, 17 pages

B. "The Impact of Object-Orientation on Software Development", A.A.R. Cockburn, **IBM Systems Journal**, 1993, Vol 32, No 3, pages 420-444.

C. "Process Automation in Software Application Development", K.D. Saracelli et al, **IBM Systems Journal**, 1993, Vol 32, No 3, pages 376-396.

14. Prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

D. Template Software Inc., product line contains the SNAP programming language and the Workflow Template. The a subset of the documentation sets for the products contains the following manuals that cover their Workflow system with RULES.

Workflow released April 3, 1997

Developing a WFT Workflow System

Using the WFT Development Environment

E. Rational Corporation's product Rational Rose C++ version 4, released in 1996.

Rational Rose C++ version 4.0 contains a *document set* containing the following documents:

- **Round-Trip Engineering with Rational Rose/C++**
- Using Rational Rose 4.0
- UML, Booch & OMT Quick Reference for Rational Rose 4.0

This product set should be viewed as the state of the technology in the mid 1990's. The Rational Rose product was eliminated from consideration because the Applicant appears to be using the term "Externalizing Rules" to mean the domain expert is the end user not the programming. Rational Rose the rules are implemented by the programmer not the domain expert. Both Martin and Template implement rules which are intended to be entered by the domain expert.

Correspondence Information

15. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to **Todd Ingberg** whose telephone number is **(703) 305-9775.** The Examiner can normally be reached on Monday, Tuesday, Thursday and Friday from 6:30 a.m. to 5:00 p.m. Until, October 22, 2001 then the Examiner's work schedule will be Monday through Thursday from 6:30 a.m. to 5:00 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the **Examiner's Supervisor**, Mark Powell is on an extended work detail, **Acting Supervisor Kevin Teska** can be

reached at **(703)305-9704.** Any response to this office action should be mailed to: **Director of Patents and Trademarks Washington, D.C. 20231** or **faxed to: (703) 308-9051**, (for formal communications intended for entry) Or: **(703) 308-1396**, (for informal or draft communications, please label "PROPOSED" or "DRAFT") **Hand-delivered** responses should be brought to **Crystal Park II, 2121 Crystal Drive Arlington, Virginia, (Receptionist located on the sixth floor).**

Todd Ingberg

September 30, 2001

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER